#2

Eur päisches
Patentamt

European
Patent Office

Office européen
des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

01107130.5

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

Hatten

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 08/10/01
LA HAYE, LE

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 01107130.5
Demande n°:

Anmeldetag:
Date of filing: 22/03/01
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
IBM Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Mechanism for dynamic extension of attributes in a content management system

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

D E S C R I P T I O N

Mechanism for Dynamic Extension of Attributes in a Content Management System

BACKGROUND OF THE INVENTION

The invention generally relates to digital content or document management systems and, more specifically, to a method for administrating or handling attributes in such a system and to a corresponding document management system.

Nowadays content management systems (CMSs) are mainly used in the Internet arena in order to manage the content of a Web site. Typically, a CMS consists of two elements: A content management application (CMA) and a content delivery application (CDA). The CMA element allows a content manager or author of the Web site to manage the creation, modification, and removal of content from the Web site without needing expertise. The CDA element uses and compiles that information to update the Web site. The features of a CMS system vary, but most include Web-based publishing, format management, revision control, and indexing, search, and retrieval.

A CMS system may also provide tools for one-to-one marketing. One-to-one marketing is the ability of a Web site to tailor its content and advertising to a user's specific characteristics using information provided by the user or gathered by the site (for example, a particular user's page sequence pattern). For example, if one visits a search engine and searches for digital cameras, the advertising banners will advertise businesses that sell digital cameras instead of businesses that sell garden products.

An additional feature of a CMS is indexing, search, and retrieval. A CMS system typically indexes all data within an organization. Individuals can then search for data using keywords, which the CMS system retrieves. Indexing is also known in file management systems thus allowing records to be accessed either sequentially, i.e. in the order they were entered, or randomly i.e. using an index. Each index defines a different ordering of the records. An employee database may have several indexes, based on the information being sought. For example, a name index may order employees alphabetically by last name, while a department index may order employees by their department. A key is specified in each index. For an alphabetical index of employee names, usually the last name field is used as the key.

A corresponding document management system is disclosed in US Pat.# 5,181,162 in which documents are represented as collections of logical components i.e. "objects" that may be combined and physically mapped onto a page-by-page layout. Stored objects are organized, accessed and manipulated through a database management system.

In the above and other known document or content management systems, documents are stored as discrete objects in an object-oriented computational environment. Stored objects are organized, accessed and manipulated through a database management system (DBMS). A DBMS is a program or part of a database product, like the IBM DB2 database management products of the present assignee, that lets one or more computer users create and access data in a database. The DBMS manages user requests and requests from other programs so that users and other programs are free from having to understand where the data is physically located on storage media and, in a multi-user system, who else may also be accessing the data. In handling user requests, the DBMS ensures the integrity of the data that is, making sure it continues to be accessible and is consistently organized as intended. It further ensures security making sure only those with access privileges can access the data.

The DBMS thus allows for a coherent, consistent encoding of object content, object attributes and inter-object relationships. At a minimum, objects contain "content", that is, basic information-bearing constituents such as text, image, voice or graphics.

Objects may also contain so-called "attributes" specifying logical or physical relationships to other objects or to a document as a whole, characteristics relating to the appearance of the content, or access restrictions. In DBMSs, attributes are also used to describe a component of the database, such as a table or a field.

In general, an attribute typically is a property or characteristic of a data item like those attributes often used to assign access rights like 'read-only'. In the well-known Hypertext Markup Language (HTML), an attribute is used as a characteristic of a page element, such as a font. An HTML user can set font attributes, such as size and color, to different values. In some programming languages an attribute is a property of an object or may be considered a container for the property of the object. For example, color might be an attribute of a text object, containing the value of "red."

In using or programming computers, generally speaking, an attribute is a changeable property or characteristic of some component of a program that can be set to different values.

Objects may also be organized according to class, permitting multiple objects to inherit the same set of characteristics and attributes. For example, a document object may be sub-classified as a patent document, and, as a consequence, all patent documents may contain the same set of content objects.

Documents themselves can be represented as objects and collected into bundles referred to generally as "folders". These folders, too, can be represented as objects.

The above described document or content management systems use distinct index classes for different kinds of documents and folders. They offer a way to search for attributes which actually exist in a searched index class. Although these systems allow for linking documents to each other, these links are not used during the search process.

In addition, the known systems do not allow for parametrizing of the underlying document management system, e.g. to supplement or extend the system with further attributes or index classes, in particular during run-time, without need to alter or change the underlying document management system itself.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a method and system for administrating attributes in a document management system which comprise an enhanced extensibility, as discussed beforehand.

It is another object to allow for an extension of attributes in a content or document management system, as discussed beforehand, during run-time of the system.

The above objects are achieved by the features of the independent claims. Advantageous embodiments are subject matter of the subclaims.

The idea underlying the invention is to provide for so-called "virtual" or "extended" attributes.

In one aspect of the invention, use of said virtual attributes enables searching for attributes in Child Index Classes and to obtain the corresponding parent index class in the search result, likewise.

According to another aspect, use of said extended attributes allows to present those search results in a Parent Class, e.g. on a graphical user interface (GUI). The extended attributes therefore support implementation of attributes having a list of possible values.

Commonly speaking, the invention allows for multiple mapping of attributes via class definitions, particularly during run-time of an underlying document management system, wherein the obtained attribute hierarchy is hidden or transparent to a user of the system. It thus enables dynamic extension of attributes during run-time of the underlying document management system.

Further, it is enabled that a graphical user interface (GUI) used in the document management system for storing and retrieving documents can be automatically adapted to a new set of attributes.

In addition, it becomes possible to assign a number of values, e.g. a list of values, to one attribute belonging to one Index Class, that can be searched for individually. This functionality is required to present, for one Index Class, single valued attributes together with multi valued attributes on the GUI wherein the presented search result is obtained via only one search request.

In the following the invention will be discussed in more detail referring to the accompanying drawings, in which

Fig. 1 is an overview block diagram for illustrating a content management system in accordance with the invention;

Fig. 2 is a schematic view of Index Class definitions according to a first embodiment of the invention; and

Fig. 3 is a similar view showing a second embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 shows a content management system 10 (CMS) having implemented an attribute administration according to the invention. The CMS 10 uses a database management system 20 DBMS) to store and retrieve multimedia content 30, e.g. text, picture, speech, video, or the like.

The CMS 10 uses a static data model for search and retrieval operation on the content. This data model can not be changed or extended once the CMS 10 contains data. The dotted line 40 illustrates the boundary of the static content description.

An e-Content Management (eCM) interface opens up the static content description using its own data model defined in an XML (eXtensible Markup Language) schema. The eCM XML schema allows for a definition of virtual and extended attributes to widen the available search space for a user. The technical details of virtual and extended attributes are discussed below.

To the user, extended and virtual attributes appear as usual attributes defined in the CMS. The eCM interface 50 translates a search request performed by the user 60 into several CMS search requests 70. The interface 50, in addition, builds up the reply for the user of the CMS replies.

The content description used by the eCM interface 50 can be changed even if the CMS already contains data. Virtual and extended attributes can be added, modified or removed at any time and the user interface can adapt to this changes if desired.

Fig. 2 shows exemplary Parent and Child IndexClass definitions in order to illustrate the concept of Virtual Attributes according to the invention. Both IndexClass definitions include for example three attributes 'A' - 'C' and 'D' . 'E'. In addition, the IndexClass Child contains an additional attribute 'V'. This attribute 'V' is to be used as a search criteria in the parent IndexClass. Therefore, the parent IndexClass has been extended by a user in order to provide an additional search criterion. This additional search criterion is called 'Virtual Attribute'.

The Virtual Attributes belong to a Child Index Class of the Content Manager and thus represent real attributes in the Child Index Class. Only in the corresponding Parent Index Class they are not existing physically, but only virtually. Thus the virtual attributes provide an additional search criterion for searches in one Index Class wherein one can search for the virtual attribute 'V' using only the corresponding Parent Index Class. The obtained search results (hits) then can be presented only as search results of the Parent Index Class and thus the performed search in the Child Index Class is hidden to the user.

If a search for a virtual attribute is to be done, the proposed

- for each element in *eList*, search folder in index class *MV_E* where *e = ???;*

- for each found folder get the child documents and add them to the resultSet r_i;

- build the intersection of all r_i.

It is noteworthy the above multiple values comprise an AND semantics.

The beforehand described extended attributes are mainly used to present search results in a Parent Class, e.g. on a GUI. They support implementation of attributes having a list of possible values.

CLAIMS

1. A method for handling attributes in a document management system which uses a database management system for storing and/or retrieving documents, wherein defining at least one child index class containing at least one additional real attribute and defining at least one corresponding parent index class containing said attribute only virtually.

2. Method according to claim 1, wherein a search for said virtual attribute is performed by the following steps:

   - Searching said child index class for said attribute;

   - for each search hit in the child index class, obtaining the corresponding parent index class;

   - building a list of said obtained parent index classes; and

   - returning said list of parent index classes with said virtual attribute.

3. A method for handling attributes in a document management system which uses a database management system for storing and/or retrieving documents, wherein at least defining a document index class having a number of attributes and a folder index class having one attribute.

4. Method according to claim 3, wherein attributes of a document are presented performing the following steps:

   - obtaining all parent folders corresponding to said folder index class;

- for each obtained parent folder, obtaining an attribute and appending said attribute to a list;

- obtaining single-valued attributes of said document and returning said single-valued attributes to said list.

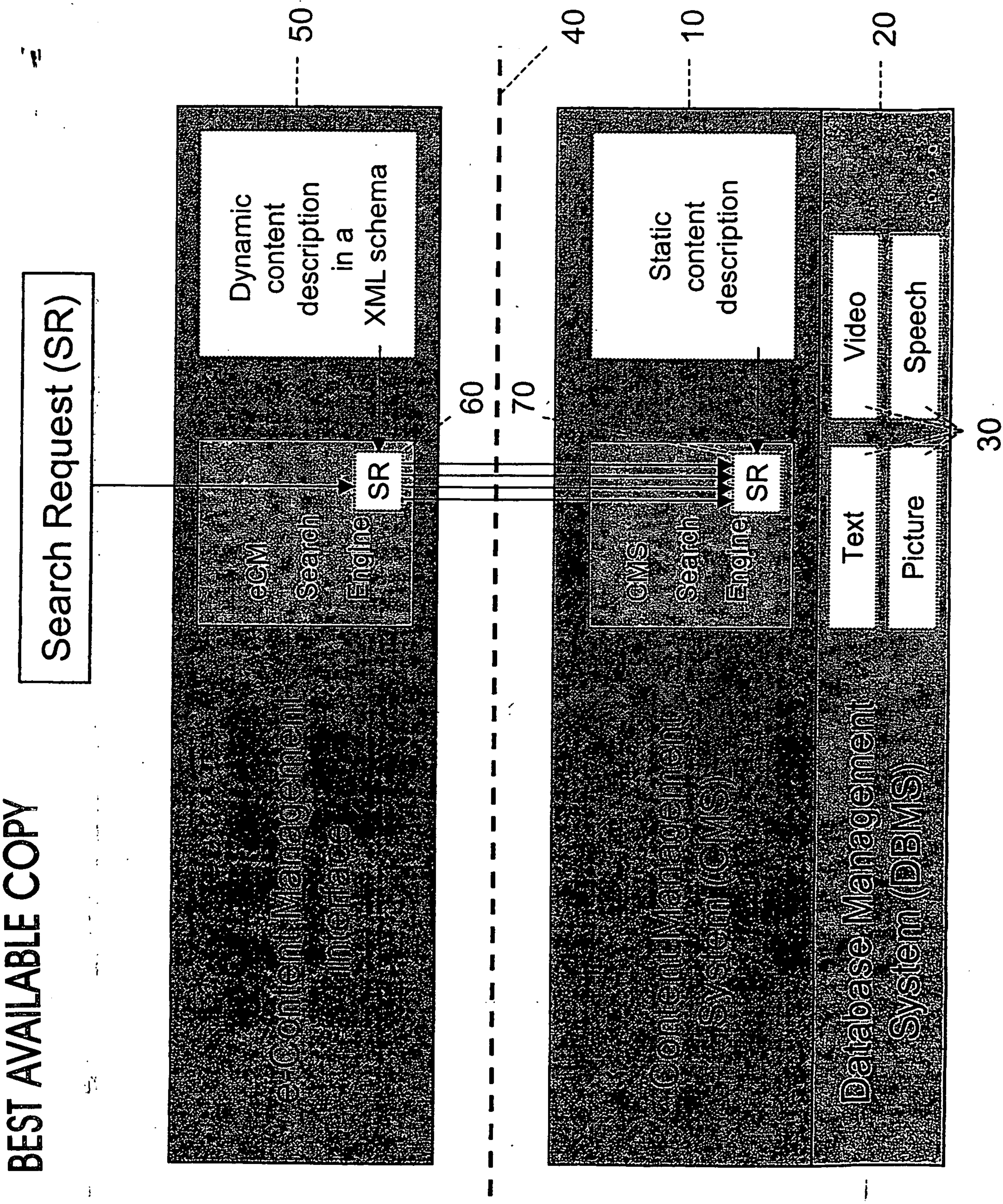
5. A document management system connected to a database management system for storing and/or retrieving documents, comprising

a document management interface connected to the document management system;

on side of the document management system, a static document description and a first search engine; and

on side of the document management interface, a dynamic document description for definition of virtual and/or extended attributes and a second search engine

6. System according to claim 5, wherein said document management interface comprises means for translating a search request into a number of document management system requests.


Search Request (SR)
50
Dynamic content description in a XML schema
eCM Search Engine
SR
e-Content Management Interface
60
70
40
10
Static content description
CMS Search Engine
SR
Content Management System (CMS)
20
Text
Video
Picture
Speech
30
Database Management System (DBMS)

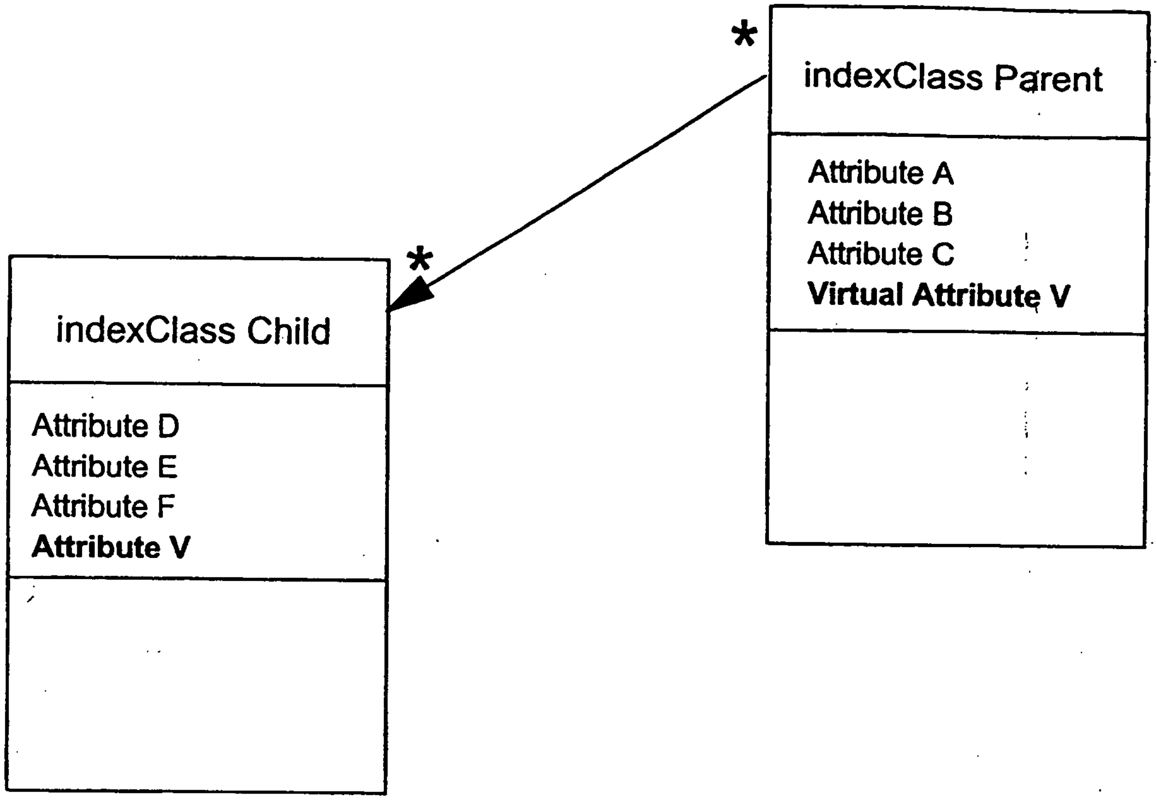

*
indexClass Parent
Attribute A
Attribute B
Attribute C
Virtual Attribute V
*
indexClass Child
Attribute D
Attribute E
Attribute F
Attribute V

Fig. 2

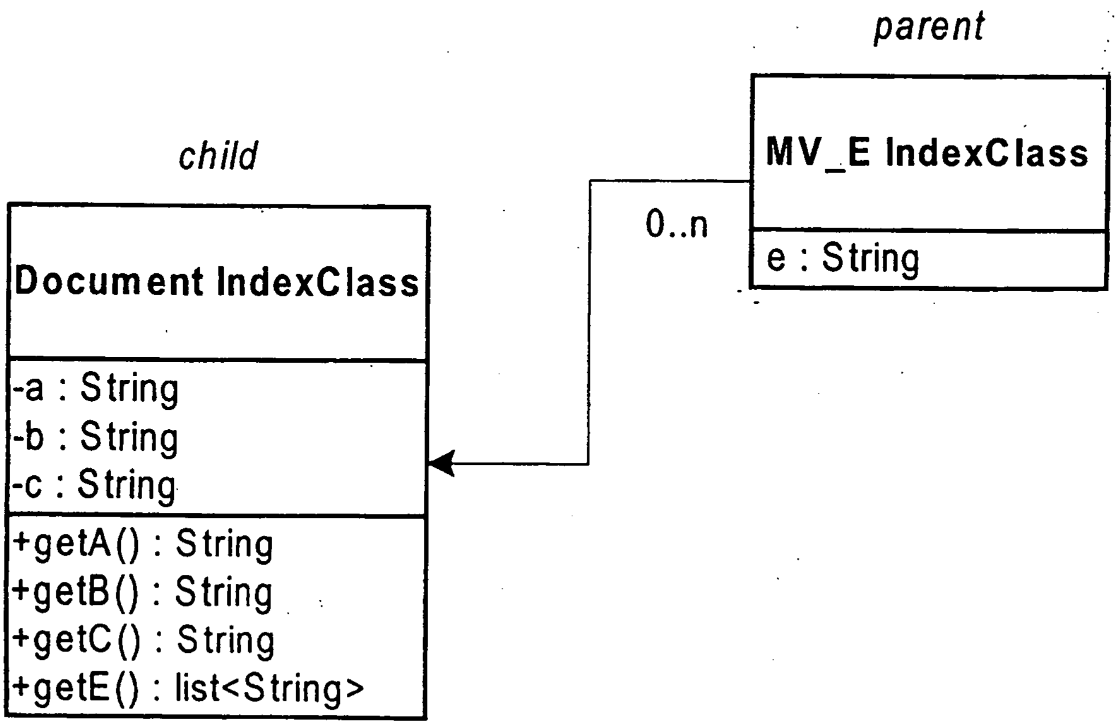

parent
MV_E IndexClass
e : String
0..n
child
Document IndexClass
-a : String
-b : String
-c : String
+getA() : String
+getB() : String
+getC() : String
+getE() : list<String>

Fig. 3

ABSTRACT

Disclosed is a mechanism for administrating attributes in a document management system. It is proposed to provide for so-called "virtual" or "extended" attributes. Use of virtual attributes enables searching for attributes in Child Index Classes and to obtain the corresponding parent index class in the search result, likewise. It therefore allows for multiple mapping of attributes via the class definitions, particularly during run-time of the underlying document management system, wherein the obtained attribute hierarchy is hidden or transparent to a user of the system. In addition, it enables dynamic extension of attributes during run-time of the underlying document management system. Use of extended attributes allows to present search results in a Parent Class, e.g. on a GUI. They support implementation of attributes having a list of possible values. Thus a graphical user interface (GUI) used in the document management system for storing and retrieving documents is automatically adapted to a new set of attributes.

(Fig. 2)